



Mimoza: web-based semantic zooming and navigation in metabolic networks

Anna Zhukova, David James Sherman

► To cite this version:

Anna Zhukova, David James Sherman. Mimoza: web-based semantic zooming and navigation in metabolic networks. BMC Systems Biology, 2015, 9, pp.10. 10.1186/s12918-015-0151-5 . hal-01123715

HAL Id: hal-01123715

<https://inria.hal.science/hal-01123715>

Submitted on 27 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SOFTWARE

Open Access

Mimoza: web-based semantic zooming and navigation in metabolic networks

Anna Zhukova* and David J Sherman

Abstract

Background: The complexity of genome-scale metabolic models makes them quite difficult for human users to read, since they contain thousands of reactions that must be included for accurate computer simulation. Interestingly, hidden similarities between groups of reactions can be discovered, and generalized to reveal higher-level patterns.

Results: The web-based navigation system Mimoza allows a human expert to explore metabolic network models in a semantically zoomable manner: The most general view represents the compartments of the model; the next view shows the generalized versions of reactions and metabolites in each compartment; and the most detailed view represents the initial network with the generalization-based layout (where similar metabolites and reactions are placed next to each other). It allows a human expert to grasp the general structure of the network and analyze it in a top-down manner

Conclusions: Mimoza can be installed standalone, or used on-line at <http://mimoza.bordeaux.inria.fr/>, or installed in a Galaxy server for use in workflows. Mimoza views can be embedded in web pages, or downloaded as COMBINE archives.

Keywords: Metabolic modeling, Visualization, Model generalization

Background

Semantic generalization of metabolic network models [1] is a theoretical method designed to aid users in understanding complex models. Generalization identifies and groups into classes biochemically similar metabolites and functionally similar reactions in the network. While we say “similar” in the commonsense way that a biologist would consider that the entities belong to the same class, we mean precisely that the two concepts are related by *is-a* relations in the corresponding ontology. For example, in a generalized model we might group all hexoses, and thus group together most hexose transporters, for a study where the differences between these transporters is not pertinent. Generalization is a kind of dimension reduction in complex models. It can also be used on several models simultaneously: a challenge in comparing models of related organisms, or in reconciling two models of the same organism, is that different curation standards may have been applied to the different

models. Generalization can bring disparate models to the same level of abstraction so that they can be compared. To explore the opportunities of model generalization, we implement it here as a practical tool that can be easily adopted and easily integrated into existing workflows.

The *zooming user interface* (ZUI) [2] paradigm has proven to be a powerful tool for representation of data at different scales. It is being adopted for various domains of applications, including cartographic [3], exploratory data visualization [4], collaborative interfaces [5] and biological data [6,7]. The challenge is how to use ZUI-based visualization for semantic generalization of metabolic networks.

Metabolic network reconstruction and infrastructure

There is a conflict between the level of detail of metabolic models needed for computer simulation and the one that can be easily analyzed by a human curator: Genome-scale metabolic models include thousands of reactions that may participate in organism’s metabolism (e.g., 2 251 reactions in the metabolic network of the bacterium *E. coli* [8], 2 352 reactions in the *yeast* 7 metabolic network model of *S. cerevisiae* [9], 7 440 reactions in *recon* 2, a global human

*Correspondence: anna.zhukova@inria.fr
Inria/Université Bordeaux/CNRS joint project-team MAGNOME, 351, cours de la Libération, F-33405 Talence, France

and Reactome [23,24] provide an extensive collection of pathways.

Models are stored and shared using established formats, such as SBML [25], SBGN-ML [26], CellML [27]. A model represented in these formats can be further enriched with the knowledge from biological databases and ontologies, e.g., ChEBI [28], Uniprot [29], by annotating elements of the models (such as metabolites, reactions) with appropriate identifiers. Further in this manuscript we will consider metabolic models in SBML format.

Although automatic model inference tools and genomic comparison methods are becoming steadily more sophisticated, they may still leave gaps in the model or add erroneous reactions. The intrinsic and extrinsic correctness of the model should be checked during the phases of analysis and curation.

Curation and analysis

The inferred draft network needs to be refined during several iterations of analysis, curation and improvement [17,30]. The goal of the *model analysis* is to verify that the model does not contain inner contradictions and errors, e.g., that the network is connected; the transport reactions between compartments are well defined;

Starting from a model for a related organism or a collection of pathways, and genomic data, they produce a draft model for the target organism. Existing metabolic models can be found in several resources, including Biomedb Database [19], BiGG [20], JWS online [21]. KEGG [22]

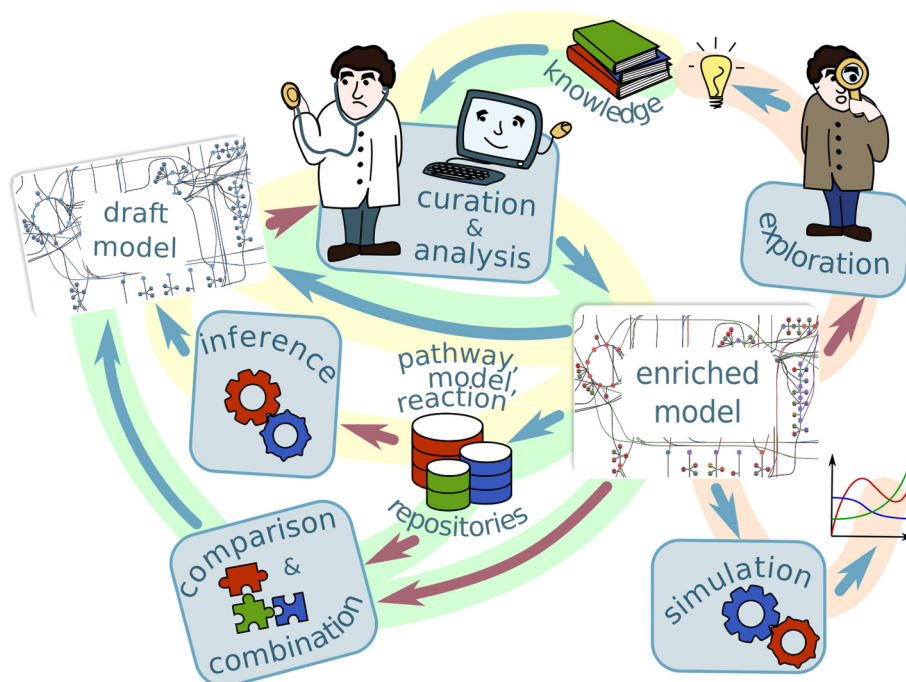


Figure 1 Metabolic modeling workflow. The processes highlighted in yellow represent the *model creation cycle*: The draft model is created by model inference tools based on models for similar organism, pathway and reaction information extracted from model repositories and pathway and reaction databases; it is then iteratively improved during the process of curation and analysis. The resulting model can in its turn be added to model repositories. The processes highlighted in red show *model usages*: simulation and knowledge-oriented exploration. The processes highlighted in green describe *comparison and combination of several models*. As the model creation cycle, they also include the curation and analysis stage. The processes represented with the red arrows require human expert's intervention, and thus need good visualization tools, ways of splitting models into modules and different levels of abstraction.

the reactions are chemically balanced, etc. Various model analysis tools, e.g. FASTGAPFILL [31] for gap filling, CellNetAnalyser [32] for finding dead ends and blocked reactions, SuBliMinaL Toolbox [17] for reaction balancing, can facilitate model analysis; but human expert's knowledge on organism's metabolism still plays an important role.

Curation is performed to ensure, first, that all of the knowledge that the experts deem pertinent is recorded in the model, and second, that the knowledge is recorded in a coherent way. The first depends on the requirements of the experts: a model for a cell factory used in an industrial process would need precise kinetics but may only require the reactions active in steady state that participate in the pathway that produces or consumes the target molecule, whereas a whole-genome model used to understand functional dependencies between genes would need to be as complete as possible but may not require reaction kinetics. The second concerns the internal consistency of what is recorded: metabolites and reactions must be annotated with ontology terms from appropriate knowledge bases,

reaction stoichiometry must be consistent, transport between compartments must be assured, and so on. Curation and analysis of models is an iterative process, ideally repeated many times to refine the draft model until the needed level of quality is achieved.

The curation by a human expert requires a means of splitting genome-scale models into smaller units that can be checked and analyzed independently. At a higher level, appropriate levels of abstraction need to be found to allow experts to compare whole genome networks. Good model visualization tools are also required.

Simulation

The improved model, created during the iterations of curation and analysis, can be used for computer simulation to obtain numerical results (see [33] for a review of simulation and flux analysis tools). We do not exploit simulation in this manuscript.

Exploration

The model can also be used for knowledge-oriented exploration to obtain new knowledge about the processes happening in the organisms' metabolism, and the relationships between them, e.g., the "redundancy" of the model: discovery of similar reactions, and alternative pathways. Means of splitting genome-scale models into smaller units, appropriate levels of abstraction and good model visualization tools are as important for model exploration task as they are for curation.

Comparison and combination

Model comparison and combination is another important task. Possible scenarios include comparison to a different model of the same organism, with potential merging into

a new, more complete, model; comparison of a model of a healthy organism to the one of a metabolism suffering from a disease to discover disease-specific metabolic adaptations. A genome-scale model can be created by combining several smaller models, describing different metabolic processes in a species [34], where model comparison is needed to detect overlaps. Such a model can be used as a draft model, and will need to undergo the analysis and curation phase. Finally, a group of models for related species can be compared and combined to produce a concise representation of their common metabolism, to study the common properties of a group, as well as the organism-specific adaptations.

There exist various software facilitating model merging, e.g., semanticSBML [35], OREMPdb [36], PathCase-SB Model Composition Tool [37], but all of them require human expert's intervention in cases when the models to be merged are incompatible or contradict to each other, as well as for better discovery of common parts. Thereby, after the creation, the combined model becomes a draft and should in its turn undergo the analysis and curation cycle.

By combining these modeling tasks into workflows, as in Figure 1, one can accomplish the modeling objectives listed above.

At least three of the aforementioned tasks (curation, exploration, comparison) require the intervention of a human expert, and thus require methods of dealing with the complexity of the models, e.g., by splitting them into smaller modules, by defining different levels of abstraction, and by visualization.

Existing visualization approaches

There exist various modeling tools for metabolic networks that also support visualization. Desktop tools include CellDesigner [38], VANTED [39], and Cytoscape [40]. They produce reasonably good visualizations of small networks (up to hundreds of reactions), but become cluttered at the genome-scale level, making the visualization unreadable.

Web-based tools allowing for metabolic network visualization are also available. JWS online [21], for example, provides a mechanism for network visualization using a force-directed layout algorithm [41,42]. It also encounters the aforementioned issues and thus is not capable of providing a readable representation for large networks.

MetDraw [43] is an online tool for genome-scale metabolic model visualization, that makes use of decomposition of the model into compartments and pathways (if the pathway information is present in the model as a *subsystem* annotation of reactions) and duplication of minor metabolites. Metabolite duplication reduces clutter, but the huge number of reactions in the compartments of some models and missing *subsystem* annotations, makes

the visualization consume too much space and do not allow a user to grasp the essential structure of the network.

Due to the huge numbers of reactions and of metabolites participating in multiple reactions, we have an uncomfortable choice between either many edge crossings in an automatic visualization of a genome-scale network, or over-duplication of various metabolites making the essential parts of the network disconnected and the visualization too large to grasp. Therefore an approach different to a simple graph layout algorithm is necessary. ZUIs, which can change the size and nature of the content displayed at different zoom levels, provide a pertinent alternative. Two main types of magnification can be considered: *geometric zooming*, in which a region of the network is enlarged; and *semantic zooming*, in which additional properties are introduced with enlargement [7].

Semantic zooming was first introduced for biological data visualization in 1988 with Zomit [6], a generic application programming interface for developing servers for zoomable navigation and visualization, and illustrated with an example of ZoomMap, a prototype browser for HuGeMap human genome database [44]. The work by Jianlu and Laidlaw [45] evaluates geometric zooming with the Google Maps interface on five examples (a gene co-regulation visualization, a gene expression heatmap viewer, a genome browser, a protein interaction network, and neural projections), and describes a positive feedback provided by both domain experts and less experienced users. Another example of a Google Maps-based ZUI is X:map [46], a genome annotation database that supports zoomable data browsing. It does not use semantic zooming, but allows for showing/hiding layers with additional information (EST and GenScan predictions).

There exist several web-based tools that include a zoomable representation of metabolic networks. Genome Projector [47] is a zoomable genome map with multiple views, including a pathway map. The pathway map is based on the Roche Biochemical Pathway wall chart available from the ExPASy proteomics server [48]. The Roche Biochemical Pathway wall chart has a large size and shows the collection of biochemically known molecules, enzymes and reactions. Genome Projector provides a geometric zooming on the map and overlay layers to highlight reactions present in the organism of interest. The list of organisms is fixed to 320 bacterial genomes. The full Roche Biochemical Pathway map with the imposed layout is always shown, but only the reactions of interest (corresponding to the chosen organism) are highlighted.

NaviCell [49] is a web environment that permits exploiting large maps of molecular interactions, including metabolic maps. It allows users to create their own maps, but does not provide a solution to the problem of huge network layout. The map creation is not fully automatic: The user must create a map in CellDesigner, export

it as an image and partly manually edit it in a graphical designer to produce intermediate views (possibly with different level of details for semantic zooming). In addition, NaviCell permits a user to split the map into submaps called modules.

Another web-based tool, the Cellular Overview [50] creates interactive diagrams for metabolic maps of organisms in the BioCyc database [51]. It is pathway-oriented, and supports only geometric zooming. Another drawback is that it does not show the compartmentalization.

The Reactome pathway database [23,24] browser provides a zommable visualization of manually curated pathways for 19 organisms. It has two semantic zoom levels: a general representation of organism's pathways (nodes represent pathways, the edges connect the related ones); and submaps showing the details of each of the pathways, including compartmentalization. Several levels of geometric zoom are available on both semantic zoom levels. Reactome is pathway-oriented. Inside each pathway the layout is imposed: reactions, metabolites, and compartments common to two organisms have the same layout in corresponding representations. On the other hand, the positions and sizes of compartments might differ between pathways of the same organism.

None of the ZUI tools for metabolic map representation described above, except for NaviCell, allow users to input their own models. Moreover, as these examples show, not only geometric zoom but also model decomposition and semantic zoom are important for multi-level visualization of huge models. At the general level, the network needs to be decomposed into several meaningful modules (such as compartments, pathways). If after such a decomposition the model remains complicated (e.g., the mitochondrial compartment of the yeast consensus model [52] containing 230 reactions), a further decomposition is required. We address these issues below by combining model generalization with a ZUI.

Implementation

Choosing zoom levels

We address the problem of large-scale metabolic model visualization by combining meaningful decomposition into modules with automatic multi-level abstraction. Decomposition is performed in the following way: The network is first split into compartments; then the model generalization method is applied to each compartment to detect the generalized modules. Thereby, the most appropriate is to adopt 3 levels of semantic zooming:

1. The most abstract level represents compartmentalization of the network, and focuses on such questions as: Are all the compartments present? Are they well connected by transport reactions?

This level shows the compartments of the model, the transport reactions between them, and other reactions happening inside the cytoplasm. If the model does not describe compartments, this level will be missing.

2. The second level shows the modules inside each of the compartments. The questions that can be addressed at this level include: Are all the reactions or more generally pathways desired by the curators present? are the input-output relations of functional modules consistent with what the expert expects from her knowledge? Does the model show organism-specific adaptations, seen in the model as shortcuts or meanders?

We use our knowledge-based generalization method to identify the modules inside the compartments. It detects similar metabolites and reactions and clusters them together to represent them as generalized metabolites and reactions with the same structure (numbers of consumed and produced metabolites). The generalized representation reveals the overall structure of the network while hiding the details. If no similar metabolites/reactions can be detected by the generalization method (due to the model structure or to missing ChEBI metabolite annotations), this level will be missing.

3. The most detailed level is intended for computer simulation and represents the inner structure of each of the modules with all the metabolites, reactions and their kinetics, stoichiometries and constraints. Our method places similar metabolites and reactions (detected at level 2) next to each other, thus simplifying the analysis of their presence.

Figure 2 shows such a 3-level representation on the example of the model of β -oxidation of fatty acids [53] in the peroxisome compartment of a yeast *Y. lipolytica*. The first level (bottom) shows the peroxisome compartment, and the transport reactions; the second level (middle) shows the generalized structure of the peroxisome; the most detailed level (top) represents the complete model, placing semantically similar metabolites and reactions next to each other.

Model generalization

The metabolic model generalization method [1], which we recall here, groups similar metabolites and reactions in the network based on its structure and the knowledge extracted from metabolite ontologies. A generalization is made specifically for a given model and is maximal with respect to the relations in the model; it respects semantic constraints such as reaction stoichiometry, connectivity, and transport between compartments; and it is performed

through a heuristic method that is efficient in practice for genome-scale models. The reader is referred to [1] for these technical details, which are beyond the scope of this article.

To make metabolite grouping semantically meaningful, an ontology describing hierarchical relationships between biochemical entities is used. Each metabolite can be generalized up to one of its ancestors in the ontology. We use the *ChEBI* ontology, as it is the *de facto* standard for metabolite annotation in metabolic networks. If a ChEBI annotation for a metabolite is not present in the model, the method attempts to automatically deduce it by comparing metabolite's name to ChEBI terms' names and synonyms.

Reactions that share the same generalized reactants and the same generalized products, are considered equivalent and are factored together into a generalized reaction.

The appropriate level of abstraction for metabolites and reactions is defined by the network itself as the most general one that satisfies two restrictions:

1. *Stoichiometry preserving restriction*: metabolites that participate in the same reaction cannot be grouped together;
2. *Metabolite diversity restriction*: metabolites that do not participate in any pair of similar reactions are not grouped together (as there is no evidence of their similarity in the network).

Overall, the generalization method is composed of three modules:

1. *Aggressive reaction grouping* based on the most general metabolite grouping (defined by ChEBI), in order to generate reaction grouping candidates;
2. *Ungrouping of some metabolites and reactions* to correct for violation of the stoichiometry preserving restriction;
3. *Ungrouping of some metabolites* (while keeping the reaction grouping intact) to correct for violation of the metabolite diversity restriction.

For instance, *(S)*-3-hydroxydecanoyl-CoA, *(S)*-3-hydroxylauroyl-CoA and *(S)*-3-hydroxytetradecanoyl-CoA have a common ancestor *hydroxy fatty acyl-CoA* in *ChEBI*. They can be grouped and generalized into *hydroxy fatty acyl-CoA*, if in the network there is no reaction whose stoichiometry would be changed by such a generalization (stoichiometry preserving restriction), and exist similar reactions that consume or produce them (metabolite diversity restriction).

The method is available as a python library [54] that operates on models in *SBML* [55] format. It takes an *SBML* file of level 2 or 3 (any version) and produces an

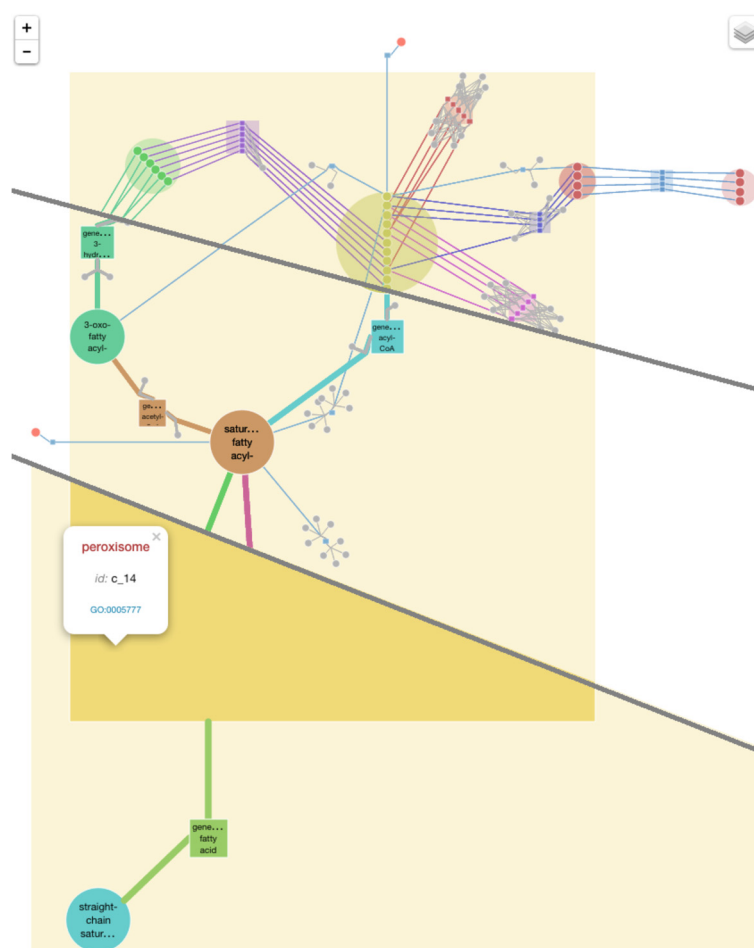


Figure 2 Three zoom levels. The most general zoom level (bottom) shows the peroxisome and a generalized transport reaction. The intermediate zoom (middle) shows the generalized processes inside the peroxisome compartment. The most detailed view (top) reveals the metabolites and reactions of the initial model.

SBML level 3 version 1 file with groups extension [56] that contains the initial model plus groups for all non-trivial similar metabolite and reaction sets (see Figure 3).

The compression that can be achieved with the model generalization method depends on the model structure and on how well the model is annotated with the ChEBI ontology (as the metabolites lacking ChEBI annotations are not generalized). Additional file 1: Table S1 shows the results of the application of the model generalization method to 269 metabolic models from Path2Model project [57]. All those models are genome-scale, the average number of reactions per model is 2879. The average compression ratio r is 1.14:

$$r = \frac{\text{number of reactions in the initial model}}{\text{number of reactions in the generalized model}} \quad (1)$$

Layers layout

To visualize a metabolic network we first represent it as a bipartite graph [58] with two disjoint sets of nodes (metabolites and reactions), and edges that connect the reactions to their substrate and product metabolites. To achieve such a representation, we implemented a converter from SBML to TLP format, that is used by the Tulip graph visualization tool [59]. TLP format stores nodes and edges of the graph, and associates each node and edge to a list of named attributes: standard ones, such as shape, size, color; and user-defined ones, such as, in our case, element type (compartment, reaction or metabolite), ChEBI identifier, group number, gene association, etc. The SBML-to-TLP converter is implemented in python, using libSBML library [60], and is available as a part of MIMOZA software.

While layout of large graphs is widely studied [61], the correspondence between the layouts of different semantic zoom levels remains a hard task. To compute the


```

<listOfSpecies>
  <species metaid= "m_s_0045" id="s_0045"
    name="(S)-3-hydroxydecanoyl-CoA [perox...]"
    compartment="c_14" ...>
    <annotation>
      <rdf:RDF xmlns:rdf="http://www.w3.org/1999/...">
        <rdf:Description rdf:about="#m_s_0045">
          <bqbiol:is><rdf:Bag>
            <rdf:li rdf:resource="ht.../chebi:28325"/>
          </rdf:Bag></bqbiol:is>
          ...
        </rdf:Description>
      </rdf:RDF>
    </annotation>
  </species>
  ...
<groups:listOfGroups>
  <groups:group metaid="m_s_gen_9" groups:id="s_gen_9"
    groups:name="hydroxy FA-CoA [pero...]"
    groups:kind="classification">
    <annotation>
      <rdf:RDF xmlns:rdf="http://www.w3.org/1999/...">
        <rdf:Description rdf:about="#m_s_gen_9">
          <bqbiol:is><rdf:Bag>
            <rdf:li rdf:resource="ht.../chebi:61902"/>
          </rdf:Bag></bqbiol:is>
          ...
        </rdf:Description>
      </rdf:RDF>
    </annotation>
    <groups:listOfMembers>
      <groups:member groups:idRef="s_0045"/>
      <groups:member groups:idRef="s_0051"/>
      <groups:member groups:idRef="s_0057"/>
      <groups:member groups:idRef="s_0054"/>
      <groups:member groups:idRef="s_0048"/>
      <groups:member groups:idRef="s_0236"/>
    </groups:listOfMembers>
  </groups:group>
  ...

```

Figure 3 Representation of a generalized model in SBML level 3 version 1 format with groups extension. The output SBML file contains the initial model (including the lists of metabolites (called *species* in SBML), reactions, etc.) plus the *listOfGroups* section that represents non-trivial quotient metabolite and reaction sets. In the figure, a group representing a quotient metabolite set of *hydroxy fatty acyl-CoAs* is shown; it includes (S)-3-hydroxydecanoyl-CoA (s_0045), (S)-3-hydroxylauoyl-CoA (s_0051), etc. Each of those metabolites was previously declared in the *listOfSpecies* section.

layout for different semantic zoom levels we combine two different approaches.

Generalized model layout

In order to lay out the sub-networks corresponding to each of the compartments after the generalization, we use a combination of standard layout algorithms provided by Tulip. We divide the compartment graph into connected components (i.e., subgraphs in which any two nodes are connected to each other by undirected paths, and which are not connected to any additional nodes in the supergraph), using a method provided by Tulip.

We then apply an appropriate layout algorithm on each of them. The results are combined together using the *Connected Component Packing* algorithm (provided by Tulip), which places the components close to each other while removing the overlaps between them.

Regarding each of the connected component sub-graphs as a directed graph (the direction of the edges is defined by the direction of the corresponding reactions; for reversible reactions edges in both directions are considered), we detect their strongly connected components (i.e., subgraphs where every vertex is reachable from every other vertex) using path-based depth-first

search algorithm [62]. Depending on the number of cycles in each strongly connected component subgraph, we choose one of the following layout algorithms, provided by Tulip:

- *Circular Layout* for the strongly connected components with less than 20 cycles (*Circular (OGDF)* [42], with $O(|E|^2)$ time and space complexity);
- For components with more cycles we use *Force-Directed Layout (FM³ (OGDF)* [63], that has the asymptotic worst-case running time of $O(|V|\log|V| + |E|)$ with linear memory requirements) to reduce the number of edge intersections.

We then represent each strongly connected component as a meta-node [59], apply a *Hierarchical Layout (Sugiyama (OGDF)* [64] algorithm (complexity of $O(|V||E|)$ in time and of $O(|V| + |E|)$ in space) on the initial connected component subgraph (that now contains no cycles), and then open the meta-nodes.

To avoid clutter we duplicate all the *minor* metabolites (*oxygen, hydrogen, water, ATP*, etc.) before applying the layout algorithms, so that there is a copy of a minor metabolite for each reaction in which it is used. We then extract a subgraph, containing all but the minor metabolites, apply the combined layout on it, and then place the minor metabolites next to the reactions in which they participate.

Generalization-based full model layout

The layout for the full model is based on the corresponding generalized model's layout. To allow zooming into the generalized model, we keep the same coordinates as in the generalized model for the minor metabolites and the ungeneralized metabolites and reactions, and place similar metabolites or reactions next to each other inside the space used by the corresponding generalized metabolites or reactions in the generalized model.

An edge in the generalized view might expand into several edges in the full-model view, for example, if it is a generalized edge connecting a generalized metabolite to a generalized reaction. The positions of the edges after such an expansion might slightly differ from the corresponding generalized one.

Node colors

A different color is assigned to each generalized metabolite/reaction; and is propagated to the corresponding metabolites/reactions of the full model. Minor metabolites are colored grey. MIMOZA's interface includes a checkbox that permits to hide/show minor metabolites.

Node sizes

The size of the nodes depends on their nature: minor metabolites are smaller than the other ones; a radius of a generalized metabolite/reaction is calculated as a sum of radii of the elements that it groups; compartment sizes are defined by the layouts of the elements inside them, so that the compartments are represented as minimal rectangles containing all the corresponding elements. All major specific (i.e., not generalized) metabolites are of the same size; as well as all specific reactions.

Relative positions of compartments

Metabolic models may include several compartments, nested into each other. For example, the *peroxisome* compartment is surrounded by its *membrane*, and contained in *cytoplasm*; the *cytoplasm* is part of the *cell*, which is surrounded by the *cell envelope*.

SBML allows to represent relative positions of the compartments in the model with an optional *outside* tag. However, it is not available in all SBML levels, nor is widely used.

To be able to visualize the compartments correctly even for the SBML models lacking this information, we infer their relative positions from the Gene Ontology (GO) [65]. We associate each compartment with a term from the *cellular component* branch of GO by using annotations in the model if they are present, or matching the compartments' names otherwise. We then use the *part_of* and *is_a* relationships between the terms in GO to infer relative compartment positions. If no term for a compartment could be found, it is placed on the outer-most level.

SBML layout

To store the calculated layout of the model elements we use the layout extension [66] of SBML. It allows to store the coordinates and sizes of the metabolites, reactions and compartments in the model. The TLP-to-SBML layout converter is implemented in python and is available as a part of MIMOZA software. If the SBML model submitted by the user contains the layout information, our software uses it for nodes' positions. Therefore, it is possible to visualize a model with MIMOZA, download the resulting SBML with layout annotations, edit it manually or with another software and then revisualize the updated version with MIMOZA.

ZUI

The zoomable interactive representation is achieved using Leaflet [67], a JavaScript library for interactive maps.

We export elements of the network graph (compartments, metabolites and reactions) as map features in GeoJSON format [68] in order to store their coordinates and metadata (e.g., ChEBI annotations for metabolites). Figure 4 shows an example of a reaction represented

```

{
  "geometry": {
    "type": "Point",
    "coordinates": [363.53, 179.46]
  },
  "type": "Feature",
  "properties": {
    "c_id": "c_14",
    "term": [
      ["YALI0E18568g"]
    ],
    "name": "acetyl-CoA acyltransferase",
    "rs": [
      ["3-oxooctadecanoyl-CoA [peroxisome]", 1],
      ["coenzyme A [peroxisome]", 1]
    ],
    "color": "#79A8C9",
    "rev": true,
    "ps": [
      ["acetyl-CoA [peroxisome]", 1],
      ["palmitoyl-CoA [peroxisome]", 1]
    ],
    "w": 1.28,
    "type": 2,
    "id": "r_0115"
  },
}

```

Figure 4 GeoJSON representation of a reaction. An SBML reaction is stored as a GeoJSON Point feature, with its layout coordinates encoded in the geometry section. The identifiers, labels and annotations, as well as the information on the reactant and product metabolites are stored as properties. The "type" property value specifies that this GeoJSON feature is a reaction.

in GeoJSON format. The TLP-to-GeoJSON converter is implemented in python and is available as a part of MIMOZA software.

The GeoJSON objects are then added as layers to the map and rendered by Leaflet into clickable elements at corresponding zoom levels. We follow SBGN Process Description language convention [69] to choose the glyphs for model elements' representation: Metabolites are drawn as circles linked by edges to the reactions where they participate; reactions are represented as squares; compartments are drawn as rectangles. On the semantic zoom levels that show compartments, the corresponding transport reactions are connected to compartments. On the more detailed zoom levels, where the metabolites inside those compartments are shown, these reactions are connected to the corresponding metabolites. When a user clicks on a map element a pop-up appears (see Figure 5) showing its name, identifier and additional information, e.g., gene associations and formulas for reactions. Two overlays allow user to show or hide minor metabolites (e.g., water, oxygen, hydrogen, etc.), and transport reactions.

Embedding

After the visualization with MIMOZA is done, we provide a link for embedding the view in another web page.

Download and distribution

One can use MIMOZA in three different ways:

1. As a standalone application. All MIMOZA code is open-source and can be downloaded from the project web page [71] and installed on a local server.
2. On the MIMOZA web server. MIMOZA web server [71] lets one test visualization for smaller SBML models, with the possibility to download the result as a COMBINE archive [72], including the SBML file with groups (to store the metabolite and reaction groupings) and layout (to store the element coordinates) extensions, GeoJSON files with the coordinates of model elements, and the HTML, CSS and JavaScript files that are needed to view the visualization in a browser.
3. As a Galaxy [73] project tool, so that generation of MIMOZA views can be included in a Galaxy workflow. The Galaxy wrapper for MIMOZA is available for download from the project web page.

Pipeline

The overall MIMOZA pipeline contains 5 steps:

1. The user submits a model in SBML format (level 2 or 3, any version) via a web form.
2. If the model does not yet contain groups, it is generalized using the model generalization method, and the resulting SBML file (level 3 version 1 with groups extension) is made available to the user.
3. The SBML file with groups of similar metabolites and reactions is converted into a Tulip graph: metabolite nodes are connected by edges to the nodes of the reactions in which they participate. The generalized metabolites and reactions form quotient nodes. The Tulip graph is split into sub-graphs corresponding to different compartments, and layout algorithms are applied to them.
4. The compartment sub-graphs are exported in GeoJSON format and rendered by the Leaflet library into an interactive map that is represented to the user.
5. The result can either be browsed on the MIMOZA web page directly, or downloaded as a COMBINE archive and embedded into a different website.

Results and discussion

To illustrate the use of MIMOZA and compare it with other available ZUI tools, we visualized the yeast consensus genome-scale metabolic network model [52]. The result can be found at <http://mimoza.bordeaux.inria.fr/yeast4>. MIMOZA automatically split the network into compartments and created a 3-level visualization for each of them.

We visualized the same model using MetDraw with no manual adjustments. The resulting SVG file^a has only one

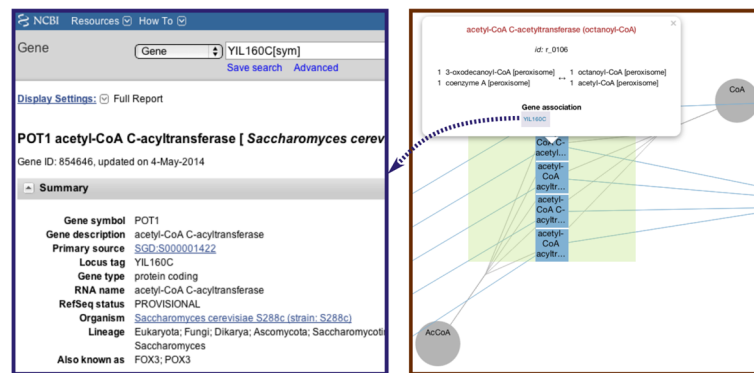


Figure 5 A reaction pop-up. (Right part) An example of a pop-up that opens when a user clicks on a reaction: It contains the information on the reaction name, identifier, reactant and product metabolites and their stoichiometries, as well as gene associations. (left part) Gene names are hyperlinks redirecting to the NCBI Gene database [70].

zoom level with lots of clutter, that does not allow one to see the structure of the network.

Cellular Overview does not allow one to visualize a model provided by a user, but has a map of metabolism of *Saccharomyces cerevisiae*.^b It has a clear non-overlapping representation of various pathways present in the model, but does not show the compartmentalization. It is not automatic and is pathway-oriented, thus is not suitable for models having no pathway metadata. The zoom-in shows additional labels but all the metabolites and reactions are present at all the levels, making the elements at the most general level very small and hard to analyze.

NaviCell does not allow to visualize an SBML model automatically. Genome Projector only contains maps for bacterial genomes and does not permit user's model input.

Neither Reactome allows users to visualize their own models, but it contains a pathway map for *Saccharomyces cerevisiae*.^c It has two semantic zoom levels: a visualization of a list of pathways present in the model, and submaps corresponding to each of them. The representation of each pathways is very clear, and has several geometric zoom levels. However, it is not always space-efficient as it contains gaps due to reactions present in other organisms but absent in *S. cerevisiae*. Another particularity is that while the positions of elements common to different organisms are conserved within a pathway, their positions might differ between different pathways of the same organism. In MIMOZA, on the contrary, the positions of the reactions and metabolites are conserved between the compartments of the same organisms; but the layout of common processes (e.g., pathways) in different organisms' visualizations might differ in the current implementation.

Table 1 summarizes the comparison of MIMOZA to other ZUI tools. MIMOZA especially targets draft models during curation, allowing one to visualize them fully

automatically and helps to analyze them in a top-down manner, starting from the general structure and going down to the details. The generalized level differentiates it from other tools, since it shows both the overall network structure and fine-grain visualization in the most detailed level, automatically placing semantically similar metabolites next to each other. MIMOZA does not depend on pathway information, automatically infers the relative compartment placement (e.g., places organelles inside the cytoplasm) and exploits a model in SBML format with ChEBI annotations for metabolites (if no annotations are present, it tries to infer them automatically based on metabolites' names).

Using generalization to compare two metabolic networks makes most sense if they have equivalent generalized nodes that can be placed in corresponding positions in the two layouts. MIMOZA currently handles this correspondence between zoom levels of the same network, but does not guarantee such correspondence when two networks are laid out independently. To meet this challenge, three strategies can be explored. The first is to use constrained layout [74], to impose the positions of key features in one network on the corresponding features of the second network. The second is also to use constrained layout, with a catalog of standard positions for common motifs in generalized maps; for example, always lay out the generalized β -oxidation of fatty acids as a 4-step cycle, with standard positions for the generalized metabolites common for all the networks that incorporate β -oxidation. The third strategy, which we are in the process of testing, is to learn a common layout by generalizing the union of the two networks. The idea is to combine the reactions into one set, run the generalization procedure on the union to fix the positions of the common features, then to build each of the layouts using only its own set of nodes. Each network layout only contains its own nodes,

Table 1 Comparison of ZUIs for metabolic models

Tool name	Imposed layout	Semantic Zoom	User's Model	Automatic Layout	Modules
Genome projector	yes	no	no	-	no
NaviCell	no	if created by user	yes	no	yes
Cellular overview	yes	no	no	-	no
Reactome	yes (same pathw. diff. org.) / no (diff. pathw. same organism)	yes	no	-	yes
Mimoza	no	yes	yes	yes	yes

but the common nodes of the two networks will be in common positions.

Finally, the API of the Leaflet framework used for the interactive navigation can be used to integrate the maps with other web-based tools, such as annotation editors or simulation software.

Mimoza is currently targeted to metabolic networks. While it can provide a geometric zooming visualization of a generic SBML model (e.g., a signaling network), the knowledge-based generalization, and therefore semantic zooming, depends on the ChEBI ontology and is intended for metabolic models. A domain-specific adaptation of the generalization method (e.g., use of a domain-specific ontology instead of ChEBI, that is targeted to metabolism) might allow Mimoza to assist in modeling of other kinds of biological networks.

Conclusions

We have implemented Mimoza, a novel software tool for automatically constructing zooming user interfaces for genome-scale metabolic models. By exploiting *model generalization*, Mimoza reduces the dimension of the model's network at outer zoom levels, and intelligently co-localizes equivalent reactions and molecular species at inner zoom levels. Consequently the biological user may efficiently navigate the high-level structure of the model; whether the goal is to understand the model or to search for errors, Mimoza exposes the important features at out zoom levels and hides the specific details in the inner ones. We provide an efficient, useful tool that is easy to adopt and, through the use of standards such as SBML and the ChEBI ontology, is easy to integrate into existing expert-centered modeling pipelines. By carefully combining model generalization with adaptive layout and open-source cartographic software, the Mimoza web server requires just a browser with Javascript. Mimoza is open source and can also be installed locally, as described on the web page, and depends on libSBML, Tulip, and Python.

Availability and requirements

Project name: Mimoza

Project home page: <http://mimoza.bordeaux.inria.fr>

Operating system(s): Platform independent

Programming language: Python, JavaScript

Other requirements: JavaScript should be enabled in the web browser. The standalone Mimoza application requires Python 2.7; libSBML-experimental ≥ 5.9 for Python with groups and layout extensions; Leaflet 0.7.3; jQuery 2.1.1 and jQuery-ui 1.10.4; Tulip ≥ 4.0 for python; and model generalization library^d.

License: CeCILL (GPL compatible)

Any restrictions to use by non-academics: no restrictions

Endnotes

^aMetDraw – <http://www.metdraw.com/metdraw/bc7df60221ba314c383b1bf6e7dad4c3056f92bb>.

^bCellular Overview – <http://biocyc.org/overviewsWeb/celOv.shtml>.

^cReactome – <http://www.reactome.org/PathwayBrowser/#SPECIES=68322&DIAGRAM=5686439>.

^dModel Generalization – <http://metamogen.gforge.inria.fr>.

Additional file

Additional file 1: Table S1. Performance of the model generalization method.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AZ and DJS conceived the study, AZ developed the software, AZ and DJS wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank Dr. Antoine Lambert for helpful discussions about using the Tulip software as a library. AZ was supported by a CORDI-S doctoral fellowship from Inria.

Received: 5 August 2014 Accepted: 6 February 2015

Published online: 26 February 2015

References

- Zhukova A, Sherman DJ. Knowledge-based generalization of metabolic models. *J Comput Biol* 2014;21(7):534–47. doi:10.1089/cmb.2013.0143.
- Bederson B, Meyer J. Implementing a zooming User Interface: experience building Pad++. *Softw Prac Exp* 1998;28(10):1101–35. doi:10.1002/(SICI)1097-024X(199808)28:10<1101::AID-SPE190>3.0.CO;2-V.
- Nivala A-M, Brewster S, Sarjakoski TL. Usability Evaluation of Web Mapping Sites. *Cartographic J* 2008;45(2):129–38. doi:10.1179/174327708X305120.
- Roberts JC. Exploratory visualization with multiple linked views In: Dykes J, MacEachren AM, Kraak M-J, editors. *Exploring Geovisualization*. Pergamon: Elsevier; 2005. p. 159–80. Chap. 8. doi:10.1016/B978-008044531-1/50426-7. <http://www.sciencedirect.com/science/article/pii/B9780080445311504267>.
- Laufer L, Halacsy P, Somlai-Fischer A. Prezi meeting: Collaboration in a zoomable canvas based environment. In: CHI '11 Extended Abstracts on Human Factors in Computing Systems. CHI EA '11. New York, NY, USA: ACM; 2011. p. 749–52. doi:10.1145/1979742.1979673. <http://doi.acm.org/10.1145/1979742.1979673>.
- Pook S, Vaysseix G, Barillot E. Zomit: biological data visualization and browsing. *Bioinformatics*. 1998;14(9):807–14.
- Hu Z, Mellor J, Wu J, Kanehisa M, Stuart JM, DeLisi C. Towards zoomable multidimensional maps of the cell. *Nat Biotechnol*. 2007;25(5):547–54. doi:10.1038/nbt1304.
- Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, et al. A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism–2011. *Mol Syst Biol*. 2011;7(1):535. doi:10.1038/msb.2011.65.
- Aung HW, Henry SA, Walker LP. Revising the Representation of Fatty Acid, Glycerolipid, and Glycerophospholipid Metabolism in the Consensus Model of Yeast Metabolism. *Ind Biotechnol*. 2013;9(4):215–28. doi:10.1089/ind.2013.0013.
- Thiele I, Swainston N, Fleming RMT, Hoppe A, Sahoo S, Aurich MK, et al. A community-driven global reconstruction of human metabolism. *Nat Biotechnol*. 2013;31(5):419–25. doi:10.1038/nbt.2488.
- von Landesberger T, Kuijper A, Schreck T, Kohlhammer J, van Wijk JJ, Fekete J-D, et al. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Comput Graph Forum*. 2011;30(6):1719–49. doi:10.1111/j.1467-8659.2011.01898.x.
- Herman I, Melancon G, Marshall MS. Graph visualization and navigation in information visualization: A survey. *IEEE Trans Vis Comput Graph*. 2000;6(1):24–43. doi:10.1109/2945.841119.
- Karp PD, Paley S, Romero P. The Pathway Tools software. *Bioinformatics*. 2002;18(Suppl 1):225–32. doi:10.1093/bioinformatics/18.suppl_1.S225.
- Agren R, Liu L, Shoaie S, Vongsangnak W, Nookaew I, Nielsen J. The RAVEN toolbox and its use for generating a genome-scale metabolic model for *Penicillium chrysogenum*. *PLoS Comput Biol*. 2013;9(3):1002980. doi:10.1371/journal.pcbi.1002980.
- Wrzodek C, Büchel F, Ruff M, Dräger A, Zell A. Precise generation of systems biology models from KEGG pathways. *BMC Syst Biol*. 2013;7:15. doi:10.1186/1752-0509-7-15.
- Pitkänen E, Jouhten P, Hou J, Syed MF, Blomberg P, Kludas J, et al. Comparative Genome-Scale Reconstruction of Gapless Metabolic Networks for Present and Ancestral Species. *PLoS Comput Biol*. 2014;10(2):1003465. doi:10.1371/journal.pcbi.1003465.
- Swainston N, Smallbone K, Mendes P, Kell D, Paton N. The SubliMinal Toolbox: automating steps in the reconstruction of metabolic networks. *J Integr Bioinformatics*. 2011;8(2):186. doi:10.2390/biecoll-jib-2011-186.
- Hamilton JJ, Reed JL. Software platforms to facilitate reconstructing genome-scale metabolic networks. *Environ Microbiol*. 2014;16(1):49–59. doi:10.1111/1462-2920.12312.
- Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, Chelliah V, et al. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst Biol*. 2010;4:92.
- Schellenberger J, Park JO, Conrad TM, Palsson BO. BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics*. 2010;11:213. doi:10.1186/1471-2105-11-213.
- Snoep JL, Olivier BG. JWS online cellular systems modelling and microbiology. *Microbiology*. 2003;149(11):3045–7.
- Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res*. 2012;40(0305-1048 (Linking)):109–14.
- Milacic M, Haw R, Rothfels K, Wu G, Croft D, Hermjakob H, et al. Annotating Cancer Variants and Anti-Cancer Therapeutics in Reactome. *Cancers*. 2012;4(4):1180–211. doi:10.3390/cancers4041180.
- Croft D. Building models using Reactome pathways as templates. *Methods Mol Biol*. 2013;1021:273–83. doi:10.1007/978-1-62703-450-0_14.
- Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*. 2003;19(4):524–31.
- van Iersel MP, Villéger AC, Czauderna T, Boyd SE, Bergmann FT, Luna A, et al. Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics (Oxford, England)*. 2012;28(15):2016–21. doi:10.1093/bioinformatics/bts270.
- Lloyd CM, Halstead MDB, Nielsen PF. CellML: its future, present and past. *Progress Biophys Mol Biol*. 2004;85(doi:10.1016/j.pbimolbio.2004.01.004):433–50.
- de Matos P, Alcántara R, Dekker A, Ennis M, Hastings J, Haug K, et al. Chemical Entities of Biological Interest: an update. *Nucleic Acids Res*. 2010;38(suppl 1):249–54. doi:10.1093/nar/gkp886.
- The UniProt Consortium. Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Res*. 2014;42(Database issue):191–8. doi:10.1093/nar/gkt1140.
- Thiele I, Palsson BO. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat Protoc*. 2010;5(1):93–121. doi:10.1038/nprot.2009.203.
- Thiele I, Vlassis N, Fleming RMT. fastGapFill: efficient gap filling in metabolic networks. *Bioinformatics*. 2014;30(17):2529–31. doi:10.1093/bioinformatics/btu321.
- Klamt S, Saez-Rodriguez J, Gilles ED. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst Biol*. 2007;1:2.
- Copeland WB, Bartley BA, Chandran D, Galdzicki M, Kim KH, Sleight SC, et al. Computational tools for metabolic engineering. *Metab Eng*. 2012;14(3):270–80.
- Schulz M, Uhlenendorf J, Klipp E, Liebermeister W. SBMLmerge, a system for combining biochemical network models. *Genome Inform Int Conf Genome Inform*. 2006;17(1):62–71.
- Krause F, Uhlenendorf J, Lubitz T, Schulz M, Klipp E, Liebermeister W. Annotation and merging of SBML models with semanticsSBML. *Bioinformatics*. 2010;26(3):421–2. doi:10.1093/bioinformatics/btp642.
- Umeton R, Nicosia G, Dewey CF. OREMPdb: a semantic dictionary of computational pathway models. *BMC Bioinformatics*. 2012;13 Suppl 4:6. doi:10.1186/1471-2105-13-S4-S6.
- Coskun SA, Cicek AE, Lai N, Dash RK, Ozsoyoglu ZM, Ozsoyoglu G. An online model composition tool for system biology models. *BMC Syst Biol*. 2013;7:88. doi:10.1186/1752-0509-7-88.
- Funahashi A, Matsuoka Y, Jouraku A, Morohashi M, Kikuchi N, Kitano H. CellDesigner 3.5: A Versatile Modeling Tool for Biochemical Networks. *Proc IEEE*. 2008;96(8):1254–65. doi:10.1109/JPROC.2008.925458.
- Rohn H, Junker A, Hartmann A, Grafarend-Belau E, Treutler H, Klapperstück M, et al. VANTED v2: a framework for systems biology applications. *BMC Syst Biol*. 2012;6(1):139. doi:10.1186/1752-0509-6-139.
- Smoot ME, Ono K, Ruscheinski J, Wang P-L, Ideker T. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*. 2011;27(3):431–2. doi:10.1093/bioinformatics/btq675. <http://bioinformatics.oxfordjournals.org/content/27/3/431.full.pdf+html>.
- Fruchterman TMJ, Reingold EM. Graph drawing by force-directed placement. *Softw Prac Exp*. 1991;21(11):1129–64. doi:10.1002/spe.4380211102.
- Tamassia R. Handbook of graph drawing and visualization (discrete mathematics and its applications). Boca Raton: Chapman & Hall/CRC; 2007.
- Jensen PA, Papin JA. MetDraw: automated visualization of genome-scale metabolic network reconstructions and high-throughput data. *Bioinformatics*. 2014. doi:10.1093/bioinformatics/btt758.
- Barillot E, Guyon F, Cussat-Blanc C, Viara E, Vaysseix G. HuGeMap: a distributed and integrated Human Genome Map database. *Nucleic Acids Res*. 1998;26(1):106–7.

45. Jianu R, Laidlaw DH. What Google Maps can do for biomedical data dissemination: examples and a design study. *BMC Res Notes*. 2013;6(1):179. doi:10.1186/1756-0500-6-179.
46. Yates T, Okoniewski MJ, Miller CJ. X:Map: annotation and visualization of genome structure for Affymetrix exon array analysis. *Nucleic Acids Res*. 2008;36(Database issue):780–6. doi:10.1093/nar/gkm779.
47. Arakawa K, Tamaki S, Kono N, Kido N, Ikegami K, Ogawa R, et al. Genome Projector: zoomable genome map with multiple views. *BMC Bioinformatics*. 2009;10(1):31. doi:10.1186/1471-2105-10-31.
48. Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel RD, Bairoch A. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*. 2003;31(13):3784–8.
49. Kuperstein I, Cohen DP, Pook S, Viara E, Calzone L, Barillot E, et al. NaviCell: a web-based environment for navigation, curation and maintenance of large molecular interaction maps. *BMC Syst Biol*. 2013;7(1):100. doi:10.1186/1752-0509-7-100.
50. Latendresse M, Karp PD. Web-based metabolic network visualization with a zooming user interface. *BMC Bioinformatics*. 2011;12(1):176.
51. Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res*. 2012;40(Database issue):742–53. doi:10.1093/nar/gkr1014.
52. Herrgård MJ, Swainston N, Dobson P, Dunn WB, Arga KY, Arvas M, et al. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nat Biotechnol*. 2008;26(10):1155–60. doi:10.1038/nbt1492.
53. Metzler DE, Metzler CM. *Biochemistry: The Chemical Reactions of Living Cells*, 2nd Edition, 2nd edn. *Biochemistry: The Chemical Reactions of Living Cells*, vol. v. 1. San Diego: Academic Press; 2001, p. 937.
54. Zhukova A, Sherman DJ. Knowledge-based metabolic model generalization library web page. <http://metamogen.gforge.inria.fr>.
55. Hucka M, Hoops S, Keating SM, Le Novère N, Sahle S, Wilkinson DJ. Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions 2008. doi:10.1038/npre.2008.2715.1.
56. Hucka M. SBML Level 3 Groups Proposal 2012. http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Groups_Proposal_Updated_%282012-06%29.
57. Büchel F, Rodriguez N, Swainston N, Wrzodek C, Czauderna T, Keller R, et al. Path2Models: large-scale generation of computational models from biochemical pathway maps. *BMC Syst Biol*. 2013;7(1):116. doi:10.1186/1752-0509-7-116.
58. Diestel R. *Graph Theory: Springer Graduate Text GTM 173: Reinhard Diestel*; 2012, p. 451. <http://identifiers.org/isbn/978-3-642-14278-9>.
59. Auber D. *Tulip – A Huge Graph Visualization Framework* In: Jünger M, Mutzel P, Farin G, Hege H-C, Hoffman D, Johnson CR, Polthier K, Rumpf M, editors. *Graph Drawing Software. Mathematics and Visualization*. Berlin Heidelberg: Springer; 2004. p. 105–26. http://dx.doi.org/10.1007/978-3-642-18638-7_5.
60. Bornstein BJ, Keating SM, Jouraku A, Hucka M. LibSBML: an API library for SBML. *Bioinformatics*. 2008;24(6):880–1.
61. Unwin A, Theus M, Hofmann H. *Graphics of large datasets: visualizing a million*. New York: Springer; 2006.
62. Gabow HN. Path-based depth-first search for strong and biconnected components. *Inf Process Lett*. 2000;74(3-4):107–14. doi:10.1016/S0020-0190(00)00051-X.
63. Hachul S, Jünger M. Large-graph layout with the fast multipole multilevel method. Technical report, Universität zu Köln, Institut für Informatik, Köln 2005. <http://e-archive.informatik.uni-koeln.de/509/>.
64. Sugiyama K, Tagawa S, Toda M. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Trans Syst Man Cybernet*. 1981;11(2):109–25. doi:10.1109/TSMC.1981.4308636.
65. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene Ontology: tool for the unification of biology. *Nat Genet*. 2000;25(1):25–9. doi:10.1038/75556.
66. Gauges R, Rost U, Sahle S, Wengler K, Bergmann FT. SBML Level 3 Layout Package Version 1 Release 1 2013. <http://identifiers.org/combine.specifications/sbml.level-3.version-1.layout.version-1.release-1> Accessed 23/10/14.
67. Agafonkin V. Leaflet - a JavaScript library for mobile-friendly maps. <http://leafletjs.com/>.
68. Butler H, Daly M, Doyle A, Gillies S, Schaub T, Schmidt C. GeoJSON Specification. <http://geojson.org/geojson-spec.html>.
69. Le Novère N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, et al. The Systems Biology Graphical Notation. *Nat Biotechnol*. 2009;27(8):735–41. doi:10.1038/nbt.1558.
70. NCBI. NCBI Gene. <http://www.ncbi.nlm.nih.gov/gene>.
71. Zhukova A, Sherman DJ. MIMOZA web page. <http://mimoza.bordeaux.inria.fr>.
72. Bergmann FT, Adams R, Moodie S, Cooper J, Glont M, Golebiewski M, et al. One file to share them all: Using the COMBINE Archive and the OMEX format to share all information about a modeling project 2014. 1407.4992.
73. Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, et al. Galaxy: a web-based genome analysis tool for experimentalists. *Current Protoc Mol Biol*. 2010;19:19–10121. doi:10.1002/0471142727.mb1910s89.
74. Böhringer K-F, Paulisch FN. Using constraints to achieve stability in automatic graph layout algorithms. In: *Proceedings of the chi — conference on human factors in computing systems*. ACM; 1990. p. 43–51. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7000>.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

